



101 028 010

Cor

101 028,010

PATENT

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

|              |   |                 |                 |
|--------------|---|-----------------|-----------------|
| Applicant:   | Steven R. Soltis                          | Examiner:       | Chirag R. Patel |
| Patent No.:  | 7,165,096                                 | Group Art Unit: | 2141            |
| Filing Date: | December 21, 2001                         | Docket No.:     | 2451            |
| Title        | Improved Storage Area Network File System |                 |                 |

Date of Deposit: 5-22-07

I hereby certify that this paper is being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450

Signature: Mary S. Keller  
Printed Name: Mary S. Keller

## REQUEST FOR CERTIFICATE OF CORRECTION

Commissioner for Patents  
Alexandria, VA 22313

Errors appear in this patent. The errors are Patent Office errors and not the Applicant. Correction thereof does not involve such changes in the patent as would constitute new matter or would require reexamination. Attached hereto is a proposed Certificate of Correction set forth on the PTO/SB/44 form.

The errors occurred in the original patent specification on page 1, Preliminary Amendment (12/21/2001), line 11-12; sheet 2 of 3, List of References cited by applicant and considered by examiner (09/19/2005), entry 9, line 1 (Other Prior Art-Non Patent Literature Documents), page 1, Preliminary Amendment (12/21/2001), lines 11-12, page 18, Specification (12/21/2001) lines 26-27 and 29-30; page 3, Claims (07/10/2006), Claim 14, line 1, and page 8, Claims (07/10/2006), Claim 42, line 2.

Certificate  
of Correction  
MAY 30 2007

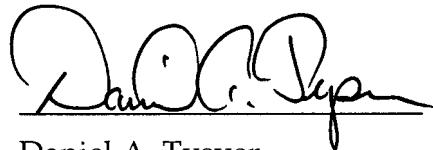
Please send the certificate to:

Daniel A. Tysver  
Beck & Tysver, P.L.L.C.  
2900 Thomas Avenue S., Suite 100  
Minneapolis, MN 55416

Since the errors are the responsibility of the Patent Office, we have not enclosed a fee. Any fee deficiency associated with this transmittal may be charged to Deposit Account 500-246.

Respectfully submitted,  
DATAPLOW, INC.  
By its attorneys:

Date: May 22, 2007



Daniel A. Tysver  
Registration No. 35,726  
Beck & Tysver, P.L.L.C.  
2900 Thomas Ave., #100  
Minneapolis, MN 55416  
Telephone: (612) 915-9634  
Fax: (612) 915-9637

**UNITED STATES PATENT AND TRADEMARK OFFICE  
CERTIFICATE OF CORRECTION**

PATENT NO. : 7,165,096

DATED : 06/16/2007

INVENTOR(S) : Steven R. Soltis

It is certified that error appears in the above-identified patent and that said Letters Patent  
is hereby corrected as shown below:

First Page, Column 1

Line 11

Above field (51) insert--

Related U.S. Application Data

Provisional application No. 60/257,446, filed on Dec. 22, 2000.---

Page 2

Column 2 (Other Publications)

Line 6 Delete "Spite" and insert -- Sprite--, therefor.

Column 1

Line 3

Above FIELD OF THE INVENTION insert--

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of Provisional application No. 60/257,446, filed on Dec. 22, 2000. - - .

Column 11

Lines 49-50 Delete "writeserialization" and insert--write-serialization--, therefor.

Line 54 Delete "readintensive" and insert --read-intensive--, therefor.

Column 20

Line 37 In Claim 14, delete "mode" and insert--inode --, therefor.

Column 23

Line 38 In Claim 42, delete "mode" and insert --inode--, therefor.

MAILING ADDRESS OF SENDER:

PATENT NO. 7,165,096

Beck & Tysver, P.L.L.C.  
2900 Thomas Avenue S., Suite 100  
Minneapolis, MN 55416

No. of additional copies



This collection of information is required by 37 CFR 1.322, 1.323, and 1.324. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1.0 hour to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Attention Certificate of Corrections Branch, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

MAY 30 2007



US007165096B2

(12) United States Patent  
Soltis(10) Patent No.: US 7,165,096 B2  
(45) Date of Patent: Jan. 16, 2007

## (54) STORAGE AREA NETWORK FILE SYSTEM

(75) Inventor: Steven R. Soltis, Rochester, MN (US)

(73) Assignee: Data Plow, Inc., Rochester, MN (US)

( \*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 788 days.

(21) Appl. No.: 10/028,010

(22) Filed: Dec. 21, 2001

## (65) Prior Publication Data

US 2002/0083120 A1 Jun. 27, 2002

## (51) Int. Cl.

G06F 15/16 (2006.01)

G06F 15/173 (2006.01)

(52) U.S. Cl. .... 709/217; 709/238; 707/1

(58) Field of Classification Search ..... 709/216,  
709/217, 238; 707/1

See application file for complete search history.

## (56) References Cited

## U.S. PATENT DOCUMENTS

|               |         |                             |
|---------------|---------|-----------------------------|
| 5,043,876 A   | 8/1991  | Terry                       |
| 5,202,971 A * | 4/1993  | Henson et al. .... 707/8    |
| 5,561,799 A * | 10/1996 | Khalidi et al. .... 707/200 |
| 5,652,913 A   | 7/1997  | Crick et al.                |
| 5,668,958 A   | 9/1997  | Bendert et al.              |
| 5,740,230 A * | 4/1998  | Vaudreuil .... 379/88.22    |
| 5,758,342 A * | 5/1998  | Gregerson .... 707/10       |
| 5,764,972 A   | 6/1998  | Crouse et al.               |
| 5,802,366 A * | 9/1998  | Row et al. .... 709/250     |
| 5,828,876 A   | 10/1998 | Fish et al.                 |
| 5,909,540 A   | 6/1999  | Carter et al.               |
| 5,931,918 A   | 8/1999  | Row et al.                  |
| 5,933,603 A   | 8/1999  | Vahalia et al.              |
| 5,978,773 A   | 11/1999 | Hudetz et al.               |

(Continued)

## OTHER PUBLICATIONS

T. Anderson, M. Dahlin, J. Neefe, D. Patterson, D. Roselli, and R. Wang, "Serverless Network File System," Proceedings of the Fifteenth ACM Symposium on Operating System Principles, 1995, The Association for Computer Machinery (ACM) Press, New York, NY USA.

(Continued)

Primary Examiner—Rupal Dharia

Assistant Examiner—Chirag R Patel

(74) Attorney, Agent, or Firm—Beck &amp; Tysver, P.L.C.

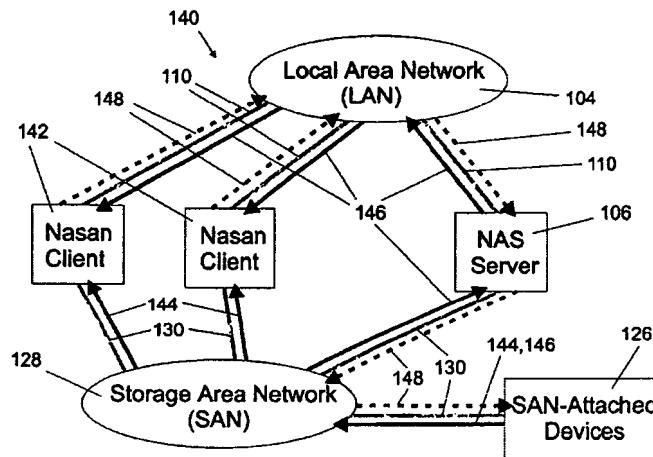
(57)

## ABSTRACT

A shared storage distributed file system is presented that provides applications with transparent access to a storage area network (SAN) attached storage device. This is accomplished by providing clients read access to the devices over the SAN and by requiring most write activity to be serialized through a network attached storage (NAS) server. Both the clients and the NAS server are connected to the SAN-attached device over the SAN. Direct read access to the SAN-attached device is provided through a local file system on the client. Write access is provided through a remote file system on the client that utilizes the NAS server. A supplemental read path is provided through the NAS server for those circumstances where the local file system is unable to provide valid data reads.

Consistency is maintained by comparing modification times in the local and remote file systems. Since writes occur over the remote file systems, the consistency mechanism is capable of flushing data caches in the remote file system, and invalidating metadata and real-data caches in the local file system. It is possible to utilize unmodified local and remote file systems in the present invention, by layering over the local and remote file systems a new file system. This new file system need only be installed at each client, allowing the NAS server file systems to operate unmodified. Alternatively, the new file system can be combined with the local file system.

61 Claims, 12 Drawing Sheets



MAY 30 2007

## U.S. PATENT DOCUMENTS

5,987,621 A 11/1999 Duso et al.  
 6,442,682 B1 \* 8/2002 Pothapragada et al. .... 713/1  
 2001/0037406 A1 \* 11/2001 Philbrick et al. .... 709/250  
 2002/0112022 A1 \* 8/2002 Kazar et al. .... 709/217

## OTHER PUBLICATIONS

M. Devarakonda, A. Mohindra, J. Simoneaux, and W. Tetzlaff, "Evaluation of Design Alternatives for a Cluster File System," 1995 USENIX Technical Conference on UNIX and Advanced Computing Systems, Jan. 1995, pp. 35-46, USENIX Association, Berkeley, CA USA.  
 G. Gibson, D. Nagle, K. Amiri, F. Chang, H. Gobioff, E. Riedel, D. Rochberg, and J. Zelenka, "File Server Scaling with Network-Attached Secure Disks," Proceedings of the ACM International Conference on Measurements and Modeling of Computer Systems, Jun. 1997, The Association for Computer Machinery (ACM) Press New York NY USA.  
 G. Gibson, D. Nagle, K. Amiri, F. Chang, E. Feinberg, H. Gobioff, C. Lee, B. Ozceri, E. Riedel, D. Rochberg, and J. Zelenka, "Filesystems for Network-Attached Secure Disks," Carnegie Mellon University Technical Report CMU-CS-97-118, Jul. 1997, Carnegie Mellon University, Pittsburgh, PA USA.  
 J. Heidemann and G. Popek, "File System Development with Stackable Layers," ACM Transaction on Computer Systems, 1994, pp. 58-89, The Association for Computer Machinery (ACM) Press, New York, NY USA.  
 Y. Khalidi and M. Nelson, "Extensible File Systems in Spring", Sun Microsystems Laboratories Technical Report TR-93-18, Sep. 1993, Sun Microsystems Laboratories, Inc., Mountain View, CA USA.  
 S. Kleiman, "Vnodes: An Architecture for Multiple File System Types in Sun UNIX," Proceedings of the Summer USENIX Conference, Jun. 1986, pp. 238-247, USENIX Association, Berkeley, CA USA.

K. Matthews, "Implementing a Shared File System on a HIPPI Disk Array," Fourteenth IEEE Symposium on Mass Storage Systems, 1995, pp. 77-88, Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ USA.

J. Ousterhout, A. Cherenson, F. Douglis, M. Nelson, and B. Welch, The Spite Network Operating System, IEEE Computer, Feb. 1988, pp. 23-36, Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ USA.

B. Pawlowski, C. Juszczak, P. Staubach, C. Smith, D. Lebel, and D. Hitz, "NFS Version 3: Design and Implementation," Proceedings of the Summer USENIX Conference, 1994, USENIX Association, Berkeley, CA.

D. Rosenthal, "Evolving the Vnode Interface," Proceedings of the Summer USENIX Conference, Jun. 1990, pp. 107-117, USENIX Association, Berkeley, CA USA.

R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and Implementation of the Sun Network File System," Proceedings of the Summer USENIX Conference, 1985, pp. 119-130, USENIX Association, Berkeley, CA USA.

M. Satyanarayanan, "Scalable, Secure, and Highly Available Distributed File Access," IEEE Computer, May 1990, pp. 9-20, Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ USA.

M. Satyanarayanan, "Code: A Highly Available File System for a Distributed Workstation Environment," Proceedings of the Second IEEE Workshop on Workstation Operating Systems, Sep. 1989, Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ USA.

S. Soltis, The Design and Implementation of a Distributed File System base on Shared Network Storage. PhD Thesis, University of Minnesota, 1997, University of Minnesota, Minneapolis, MN USA.

D. Wiltzus and K. Minuzzo, "Network-attached peripherals (NAP) for HPSS/SIOF," Lawrence Livermore National Laboratory Technical Report, 1995. Available at [http://www.llnl.gov/liv\\_comp/siof/siof\\_nap.html](http://www.llnl.gov/liv_comp/siof/siof_nap.html), Livermore, CA USA.

\* cited by examiner

MAY 30 2007

**STORAGE AREA NETWORK FILE SYSTEM****FIELD OF THE INVENTION**

The present invention relates general to computer file systems. More specifically, the present invention involves a distributed file system that transfers data using both network attached storage (NAS) and storage area network (SAN) protocols.

**BACKGROUND OF THE INVENTION****File Systems**

The term "file system" refers to the system designed to provide computer application programs with access to data stored on storage devices in a logical, coherent way. File systems hide the details of how data is stored on storage devices from application programs. For instance, storage devices are generally block addressable, in that data is addressed with the smallest granularity of one block; multiple, contiguous blocks form an extent. The size of the particular block, typically 512 bytes in length, depends upon the actual devices involved. Application programs generally request data from file systems byte by byte. Consequently, file systems are responsible for seamlessly mapping between application program address-space and storage device address-space.

File systems store volumes of data on storage devices. The term "volume" refers to the collection of data blocks for one complete file system instance. These storage devices may be partitions of single physical devices or logical collections of several physical devices. Computers may have access to multiple file system volumes stored on one or more storage devices.

File systems maintain several different types of files, including regular files and directory files. Application programs store and retrieve data from regular files as contiguous, randomly accessible segments of bytes. With a byte-addressable address-space, applications may read and write data at any byte offset within a file. Applications can grow files by writing data to the end of a file; the size of the file increases by the amount of data written. Conversely, applications can truncate files by reducing the file size to any particular length. Applications are solely responsible for organizing data stored within regular files, since file systems are not aware of the content of each regular file.

Files are presented to application programs through directory files that form a tree-like hierarchy of files and subdirectories containing more files. Filenames are unique to directories but not to file system volumes. Application programs identify files by pathnames comprised of the filename and the names of all encompassing directories. The complete directory structure is called the file system namespace. For each file, file systems maintain attributes such as ownership information, access privileges, access times, and modification times.

File systems often utilize the services of operating system memory caches known as buffer caches and page caches. These caches generally consist of system memory buffers stored in volatile, solid-state memory of the computer. Caching is a technique to speed up data requests from application programs by saving frequently accessed data in memory for quick recall by the file system without having to physically retrieve the data from the storage devices. Caching is also useful during file writes; the file system may write data to the memory cache and return control to the appli-

cation before the data is actually written to non-volatile storage. Eventually, the cached data is written to the storage devices.

The state of the cache depends upon the consistency between the cache and the storage devices. A cache is "clean" when its contents are exactly the same as the data stored on the underlying storage devices. A cache is "dirty" when its data is newer than the data stored on storage devices; a cache becomes dirty when the file system has written to the cache, but the data has not yet been written to the storage devices. A cache is "stale" when its contents are older than data stored on the storage devices; a cache becomes stale when it has not been updated to reflect changes to the data stored on the storage devices.

In order to maintain consistency between the caches and the storage devices, file systems perform "flush" and "invalidate" operations on cached data. A flush operation writes dirty cached data to the storage devices before returning control to the caller. An invalidation operation removes stale data from the cache without invoking calls to the storage devices. File systems may flush or invalidate caches for specific byte-ranges of the cached files.

Many file systems utilize data structures called inodes to store information specific to each file. Copies of these data structures are maintained in memory and within the storage devices. Inodes contain attribute information such as file type, ownership information, access permissions, access times, modification times, and file size. Inodes also contain lists of pointers that address data blocks. These pointers may address single data blocks or address an extent of several consecutive blocks. The addressed data blocks contain either actual data stored by the application programs or lists of pointers to other data blocks. With the information specified by these pointers, the contents of a file can be read or written by application programs. When an application programs write to files, data blocks may be allocated by the file system. Such allocation modifies the inodes.

Additionally, file systems maintain information, called "allocation tables", that indicate which data blocks are assigned to files and which are available for allocation to files. File systems modify these allocation tables during file allocation and de-allocation. Most modern file systems store allocation tables within the file system volume as bitmap fields. File systems set bits to signify blocks that are presently allocated to files and clear bits to signify blocks available for future allocation.

The terms real-data and metadata classify application program data and file system structure data, respectively. In other words, real-data is data that application programs store in regular files. Conversely, file systems create metadata to store volume layout information, such as inodes, pointer blocks, and allocation tables. Metadata is not directly visible to applications. Metadata requires a fraction of the amount of storage space that real-data occupies and has significant locality of reference. As a result, metadata caching drastically influences file system performance.

Metadata consistency is vital to file system integrity. Corruption of metadata may result in the complete destruction of the file system volume. Corruption of real-data may have bad consequences to users but will not affect the integrity of the whole volume.

**I/O Interfaces**

I/O interfaces transport data among computers and storage devices. Traditionally, interfaces fall into two categories: channels and networks. Computers generally communicate with storage devices via channel interfaces. Channels predictably transfer data with low-latency and high-bandwidth

11

The I/O interface links 130 connect to the SAN 128, which consists of network components such as routers, switches, and hubs. The SAN 128 may also include components that perform storage virtualization, caching, and advanced storage management functions. The SAN devices 126 are block and object-addressable, non-volatile storage devices. The SAN devices 126 may be part of a SAN appliance 136 or dedicated storage devices attached to the SAN 128.

The primary read data-path 144 of Nasan is similar to the read data-path 132 of prior art SAN environments 120, whereas the secondary read data-path 146 is similar to the read data-path 114 of prior art NAS environments 100. The majority of read transfers take place over the primary data-path 144, which passes from the SAN devices 126, through the SAN 128, directly to the Nasan clients 142. The primary data-path 144 takes full advantage of high-speed SAN protocols. However, some read transfers follow the secondary data-path 146 and pass from the SAN-attached devices 126, through the NAS server 106, across the LAN 104, en-route to the Nasan clients 142. The state of the Nasan environment 140 dictates whether the primary data-path 144 or the secondary data-path 146 is used for read transfers.

The write data-path 148 of Nasan is similar to the write data-path of prior art NAS 116 with the difference being the Nasan write data-path 148 also includes the SAN 128. The write data-path 148 begins at the Nasan clients 142 and passes through the LAN 104 to the NAS server 106. The server 106, in turn, writes across the SAN 128 to the SAN-attached devices 126.

Due to high-speed SAN reads 144, the Nasan file system significantly exceeds the file sharing performance and scalability of prior art NAS solutions. Although Nasan write performance is similar to prior art NAS write performance, Nasan reads are often ten times faster. Because read operations generally outnumber writes five to one, the performance improvement made to reads dramatically increases overall system throughput. Furthermore, by offloading reads from the NAS servers 106, the Nasan file system substantially reduces server 106 workloads. With reduced workloads, servers 106 exhibit shorter response times, sustain more simultaneous file transfers, and support considerably larger throughputs than servers 106 supporting traditional NAS 100.

The Nasan file system transfers read requests across the high-speed SAN 128 while serializing writes through a central NAS server 106. This serialization leads to write transfer rates that are slower than reads; however, ~~writeserialization~~ facilitates extremely low-latency consistency management. Low-latency consistency enables Nasan clients 142 to efficiently transfer files of all sizes. Therefore, the Nasan file system is a general-purpose solution for ~~readintensive~~ workloads.

#### Nasan Layering

One embodiment of the Nasan file system utilizes a two-tiered layering scheme. Nasan software occupies the upper level, while non-modified local and remote file systems comprise the lower. The Nasan layer provides a management framework that facilitates data consistency and routes file requests to the appropriate lower level file system. All remaining file management functionality is derived from these lower layer file systems.

Referring to FIG. 5, application programs 150 running on the Nasan client 142 make file requests to the Nasan file system software layer 152. Nasan software 152 redirects read requests to the either the local file system level 154 or

12

the client-side remote file system layer 156 and redirects write requests to the remote file system layer 156. These lower layer file systems conduct the actual data management, transport, and storage.

The local file system 154 of the client provides the primary read data-path 144 for Nasan transfers. Because the clients 142 do not directly modify the volume stored on the SAN devices 126, Nasan software 152 maintains low-latency consistency by simply invalidating stale caches of the local file system layer 154.

The remote file system facilitates the secondary read data-path 146 as well as write access to files managed by the NAS server 106. The Nasan client 142 passes file requests to the client-side remote file system layer 156. In turn, the remote file system 156 on the client 142 transmits these requests via NAS protocols to the server-side remote file system layer 158 on the server 106. The NAS server 106 completes the requests by reading data from or writing data through the local file system 155 of the server 106 to volumes stored on SAN-attached devices 126. Write-serialization, through the NAS server 106, enables low-latency consistency.

#### Components of the Preferred Embodiment

The components and protocols that form the environment 140 of the present invention range in price, performance, and compatibility. In the preferred embodiment, the interface links 110,130 that connect to the LAN 104 and to the SAN 128 may include Ethernet, InfiniBand, and Fibre Channel. Over these links 110,130 run a number of different network and channel protocols, including Internet Protocol (IP), SCSI-3, Virtual Interface (VI), iSCSI, FCIP, and iFCP. The NAS protocols used by the remote file system 156,158 include Network File System (NFS), Server Message Block (SMB), and Common Internet File System (CIFS). The present invention is not limited to these specific components and protocols.

#### Local File System Consistency

In general, local file systems perform extensive metadata and real-data caching. The only consistency management typically required of local file systems is periodic updates to on-disk data structures. Cached data is never invalidated because on-disk data is always assumed to be older than cached data.

Within a Nasan environment 140, the NAS server 106 has read-write access to the local file system volume stored on SAN-attached disks 126, while Nasan clients 142 have read-only access to this volume. Because the client local file systems 154 and the server local file systems 155 may not be designed to support SAN environments with multiple computers, Nasan software 152 must explicitly maintain data consistency between storage devices 126 and caches of the client local file system 154.

#### Consistency Between Local and Remote File System Layers

Local 154 and remote 156 file systems utilize separate caches within client 142 main memories. After file writes, the remote file system 156 cache contains newer data than the local file system 154 cache. Nasan software 152 makes the local file system 154 cache consistent with the Nasan environment 140 by explicitly invalidating stale data within the cache.

Nasan software 152 has the option to read from the local file system 154 or the remote file system 156. When reading from the primary data-path 144, Nasan software 152 first determines if data is being cached by the client-side remote file system 156. If data is cached, Nasan software 152 flushes the remote file system 156 cache and invalidates the

11/11/2007

19

locked, the Nasan client 142 reads the allocation tables from the SAN devices 126, modifies the allocation tables, writes the tables to the SAN devices 126, and then releases the file lock.

At step 418, the file is fully allocated for the request range. The Nasan write function writes the real-data to the SAN-attached devices 126 via the SAN write data-path 326. Once this real-data write completes, at step 420, the modified on-disk inode is written by the client 142 to the SAN-attached devices 126 and the file lock is released by issuing an unlock request to the client-side remote file system 156. The remote file system 156 passes the unlock request to the server 106 which forwards the unlock request to the server-side Nasan file system 324. After the file lock is released, the Nasan write operation completes.

The invention is not to be taken as limited to all of the details thereof as modifications and variations thereof may be made without departing from the spirit or scope of the invention. For instance, the present invention was described and shown with the SAN and LAN networks appearing as separate, physical networks. However, as is well known in the prior art, it is possible to send SAN protocols and LAN protocols over the same physical network. The two networks are distinguishable by the protocols that are used to communicate between nodes on the network. In addition, although it is not shown in the drawings, it would be possible to use a client computer in the present invention as a file server that serves file requests from other computers. These other computers would likely have no access to the storage area network, but would have the ability to send file requests to the client computer of the present invention over a local area network. Because many such modifications and variations are present, the scope of the present invention is not to be limited to the above description, but rather is to be limited only by the following claims.

What is claimed is:

1. A file system on a computer handling file read and file write requests to a SAN-attached storage device comprising:

- a) a local component that communicates with the SAN-attached storage device over a storage area network and that interprets metadata stored on the SAN-attached storage device;
- b) a NAS server that communicates with the SAN-attached storage device over a storage area network;
- c) a remote component that communicates with the NAS server over a local area network;
- d) an upper level component that communicates with application programs, the upper level component submitting all file write requests to the remote component and submitting at least some file read requests to the local component.

2. The file system of claim 1), wherein each of the components are separate file systems in which the upper level component file system is layered above the local component file system and the remote component file system.

3. The file system of claim 2) utilizing an installable file system interface to facility layering between the file systems.

4. The file system of claim 3) is the Virtual File System interface.

5. The file system of claim 2), wherein the remote component file system utilizes a protocol chosen from among the following set: Network File System, Server Message Block, and Common Internet File System.

6. The file system of claim 1), wherein the upper level component and the local component are merged into a single

20

file system, and further wherein the remote component is a separate file system layered under the single file system containing the upper level and local components.

7. The file system of claim 1), wherein the upper level component submits all read requests to the local component.

8. The file system of claim 1), wherein the upper level component submits read requests above a certain size to the local component, and read requests below the certain size to the remote component.

9. The file system of claim 1), wherein the upper level component submits read requests to the local component if the file size is above a certain size, and the upper level component submits read requests to the remote component if the file size is below the certain size.

10. The file system of claim 1), wherein the upper level component submits all read request to the local component except where the local component is not capable of properly retrieving data requested in the read request, in which case the upper level component submits the read request to the remote component.

11. The file system of claim 10), wherein the upper level component determines whether the local component is capable of properly retrieving data requested in the read request by comparing modification times for a file indicated in the read request as retrieved from the remote component and the local component.

12. The file system of claim 11), further wherein the upper level component determines whether the local component is capable of properly retrieving data requested in the read request by comparing modification times for a directory indicated in the read request as retrieved from the remote component and the local component.

13. The file system of claim 12), wherein the directory modification times are compared during a lookup function, with the results stored in an inode structure for the higher-level component.

14. The file system of claim 13), wherein the [mode] structure for the higher-level component includes the following:

- a) a handle pointing to a file vnode for the remote component;
- b) a handle pointing to a file vnode for the local component;
- c) a remote modification time indicating the modification time returned by the remote component; and
- d) a local modification time indicating the modification time returned by the local component.

15. The file system of claim 1), wherein the file system is capable of handling additional file requests, with file requests that alter data on the SAN-attached storage device being treated similar to the write requests, and file requests that do not alter data on the SAN-attached storage device being treated similar to the read requests.

16. The file system of claim 1), further comprising a file server component capable of receiving and responding to file requests from other computers that are connected to the local area network but not connected to the storage area network.

17. A network of connected computing devices comprising:

- a) a local area network;
- b) a storage area network;
- c) a SAN-attached device attached to the storage area network;
- d) a server computer attached to the local area network and the storage area network; the server computer receiving file requests across the local area network and

AT 302007

iii) determining whether file requests that do not modify the SAN-attached device can be fulfilled through the local file system;

iv) treating file requests that do not modify the SAN-attached device and can be fulfilled through the local file system as local requests; and

v) treating the remaining file requests as remote requests.

39. The method of claim 37), wherein the step of determining whether file requests can be fulfilled through the local file system further comprises:

(1) determining a file involved in the file requests;  
 (2) retrieving a modification time for the file from the local file system;

(3) retrieving a modification time for the file from the remote file system; and

(4) comparing the local and remote modification times.

40. The method of claim 39), wherein the step of determining whether file requests can be fulfilled through the local file system further comprises:

(5) invalidating the local file system cache when the two modification times are not identical.

41. The method of claim 39), wherein the step of determining whether file requests can be fulfilled through the local file system further comprises:

(6) obtaining a new modification time from the local file system after the local file system cache has been invalidated;

(7) comparing the new local modification time with the remote modification time in a second comparison;

(8) treating the file request as a local request if the second comparison finds the modification times to be identical; and treating the file request as a remote request if the second comparison does not find the modification times to be identical.

42. The method of claim 41), further comprising the step of performing a lookup function to create an inode for the file, the [mode] having a remote handle pointing to a remote vnode for the file and a local handle pointing to a local vnode for the file, the remote and local vnodes being used to identify the file in the remote and local file systems, respectively.

43. The method of claim 42), wherein the step of performing a lookup function compares modification times for the directory containing the file in both the remote and local file systems to determine whether the local file system access should be allowed.

44. The method of claim 43), wherein a lookup routine is performed for the file for both the local and remote file systems, and further wherein, if the remote file system does not find the file, a file not found result is returned to the application.

45. The method of claim 44), wherein if the local file system does not find the file, the remote file system is used to service file requests.

46. A network of connected computing devices comprising:

- a) a local area network;
- b) a storage area network;
- c) a SAN-attached device attached to the storage area network;
- d) a server computer attached to the local area network and the storage area network; the server computer receiving file requests across the local area network and further storing and retrieving data on the SAN-attached device via the storage area network; and

e) at least one client computer attached to the local area network and the storage area network; the client computer having:

i) a remote component in communication with and making file requests to the server computer over the local area network;

ii) a local component in communication with and making metadata and real-data requests to the SAN-attached device over the storage area network;

iii) an upper level component serving file requests from an application program operating on the client computer, the upper level component dividing the file requests from the application program between the remote component and the local component.

15 47. The network of claim 46), wherein the upper level component submits all file requests having data sizes above a certain size to the SAN-attached device over the storage area network via the local component and all file requests having data sizes below a certain size to the server computer over the local area network via the remote component.

16 48. The network of claim 46), wherein the upper level component submits all file requests to the SAN-attached device over the storage area network via the local component if the file size is above a certain size, and the upper level

25 component submits all file requests to the server computer over the local area network via the remote component if the file size is below the certain size.

49. The network of claim 46), wherein file requests comprise write requests and read requests, and the upper 30 level component submits all write requests to the server computer over the local area network via the remote component, and at least some of the read requests to the SAN-attached device over the storage area network via the local component.

50. The network of claim 49), wherein the upper level component submits all read requests having data sizes above a certain size to the SAN-attached device via the local component, and further wherein the upper level component submits all read requests having data sizes below a certain 40 size to the server computer via the remote component.

51. The network of claim 49), wherein the upper level component submits all read requests to the SAN-attached device via the local component if the file size is above a certain size, and further wherein the upper level component submits all read requests to the server computer via the remote component if the file size is below the certain size.

52. The network of claim 49), wherein the upper level component submits all read requests through the local component except where the local file system is not capable of properly retrieving data requested in the read request, in which case the upper level component submits the read request to the remote component.

53. The network of claim 52), wherein the upper level component determines whether the local component is capable of properly retrieving data requested in the read requests by comparing modification times received from the local component with modification times received from the remote component.

54. The network of claim 49), further comprising a 60 plurality of additional client computers, wherein, in relation to the client computers, only the server computer is granted write access to the SAN-attached device and further wherein all write requests from the client computers are routed via the remote component in the client computers to the server computer.

55. The file system of claim 46), wherein the client computer further has a file server component that receives 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000 3100 3200 3300 3400 3500 3600 3700 3800 3900 4000 4100 4200 4300 4400 4500 4600 4700 4800 4900 5000 5100 5200 5300 5400 5500 5600 5700 5800 5900 6000 6100 6200 6300 6400 6500 6600 6700 6800 6900 7000 7100 7200 7300 7400 7500 7600 7700 7800 7900 8000 8100 8200 8300 8400 8500 8600 8700 8800 8900 9000 9100 9200 9300 9400 9500 9600 9700 9800 9900 10000 10100 10200 10300 10400 10500 10600 10700 10800 10900 11000 11100 11200 11300 11400 11500 11600 11700 11800 11900 12000 12100 12200 12300 12400 12500 12600 12700 12800 12900 13000 13100 13200 13300 13400 13500 13600 13700 13800 13900 14000 14100 14200 14300 14400 14500 14600 14700 14800 14900 15000 15100 15200 15300 15400 15500 15600 15700 15800 15900 16000 16100 16200 16300 16400 16500 16600 16700 16800 16900 17000 17100 17200 17300 17400 17500 17600 17700 17800 17900 18000 18100 18200 18300 18400 18500 18600 18700 18800 18900 19000 19100 19200 19300 19400 19500 19600 19700 19800 19900 20000 20100 20200 20300 20400 20500 20600 20700 20800 20900 21000 21100 21200 21300 21400 21500 21600 21700 21800 21900 22000 22100 22200 22300 22400 22500 22600 22700 22800 22900 23000 23100 23200 23300 23400 23500 23600 23700 23800 23900 24000 24100 24200 24300 24400 24500 24600 24700 24800 24900 25000 25100 25200 25300 25400 25500 25600 25700 25800 25900 26000 26100 26200 26300 26400 26500 26600 26700 26800 26900 27000 27100 27200 27300 27400 27500 27600 27700 27800 27900 28000 28100 28200 28300 28400 28500 28600 28700 28800 28900 29000 29100 29200 29300 29400 29500 29600 29700 29800 29900 30000 30100 30200 30300 30400 30500 30600 30700 30800 30900 31000 31100 31200 31300 31400 31500 31600 31700 31800 31900 32000 32100 32200 32300 32400 32500 32600 32700 32800 32900 33000 33100 33200 33300 33400 33500 33600 33700 33800 33900 34000 34100 34200 34300 34400 34500 34600 34700 34800 34900 35000 35100 35200 35300 35400 35500 35600 35700 35800 35900 36000 36100 36200 36300 36400 36500 36600 36700 36800 36900 37000 37100 37200 37300 37400 37500 37600 37700 37800 37900 38000 38100 38200 38300 38400 38500 38600 38700 38800 38900 39000 39100 39200 39300 39400 39500 39600 39700 39800 39900 40000 40100 40200 40300 40400 40500 40600 40700 40800 40900 41000 41100 41200 41300 41400 41500 41600 41700 41800 41900 42000 42100 42200 42300 42400 42500 42600 42700 42800 42900 43000 43100 43200 43300 43400 43500 43600 43700 43800 43900 44000 44100 44200 44300 44400 44500 44600 44700 44800 44900 45000 45100 45200 45300 45400 45500 45600 45700 45800 45900 46000 46100 46200 46300 46400 46500 46600 46700 46800 46900 47000 47100 47200 47300 47400 47500 47600 47700 47800 47900 48000 48100 48200 48300 48400 48500 48600 48700 48800 48900 49000 49100 49200 49300 49400 49500 49600 49700 49800 49900 50000 50100 50200 50300 50400 50500 50600 50700 50800 50900 51000 51100 51200 51300 51400 51500 51600 51700 51800 51900 52000 52100 52200 52300 52400 52500 52600 52700 52800 52900 53000 53100 53200 53300 53400 53500 53600 53700 53800 53900 54000 54100 54200 54300 54400 54500 54600 54700 54800 54900 55000 55100 55200 55300 55400 55500 55600 55700 55800 55900 56000 56100 56200 56300 56400 56500 56600 56700 56800 56900 57000 57100 57200 57300 57400 57500 57600 57700 57800 57900 58000 58100 58200 58300 58400 58500 58600 58700 58800 58900 58900 59000 59100 59200 59300 59400 59500 59600 59700 59800 59900 60000 60100 60200 60300 60400 60500 60600 60700 60800 60900 61000 61100 61200 61300 61400 61500 61600 61700 61800 61900 62000 62100 62200 62300 62400 62500 62600 62700 62800 62900 63000 63100 63200 63300 63400 63500 63600 63700 63800 63900 64000 64100 64200 64300 64400 64500 64600 64700 64800 64900 65000 65100 65200 65300 65400 65500 65600 65700 65800 65900 66000 66100 66200 66300 66400 66500 66600 66700 66800 66900 67000 67100 67200 67300 67400 67500 67600 67700 67800 67900 68000 68100 68200 68300 68400 68500 68600 68700 68800 68900 68900 69000 69100 69200 69300 69400 69500 69600 69700 69800 69900 70000 70100 70200 70300 70400 70500 70600 70700 70800 70900 71000 71100 71200 71300 71400 71500 71600 71700 71800 71900 72000 72100 72200 72300 72400 72500 72600 72700 72800 72900 72900 73000 73100 73200 73300 73400 73500 73600 73700 73800 73900 74000 74100 74200 74300 74400 74500 74600 74700 74800 74900 75000 75100 75200 75300 75400 75500 75600 75700 75800 75900 76000 76100 76200 76300 76400 76500 76600 76700 76800 76900 76900 77000 77100 77200 77300 77400 77500 77600 77700 77800 77900 77900 78000 78100 78200 78300 78400 78500 78600 78700 78800 78900 78900 79000 79100 79200 79300 79400 79500 79600 79700 79800 79900 80000 80100 80200 80300 80400 80500 80600 80700 80800 80900 80900 81000 81100 81200 81300 81400 81500 81600 81700 81800 81900 81900 82000 82100 82200 82300 82400 82500 82600 82700 82800 82900 82900 83000 83100 83200 83300 83400 83500 83600 83700 83800 83900 83900 84000 84100 84200 84300 84400 84500 84600 84700 84800 84900 84900 85000 85100 85200 85300 85400 85500 85600 85700 85800 85900 85900 86000 86100 86200 86300 86400 86500 86600 86700 86800 86900 86900 87000 87100 87200 87300 87400 87500 87600 87700 87800 87900 87900 88000 88100 88200 88300 88400 88500 88600 88700 88800 88900 88900 89000 89100 89200 89300 89400 89500 89600 89700 89800 89900 90000 90100 90200 90300 90400 90500 90600 90700 90800 90900 90900 91000 91100 91200 91300 91400 91500 91600 91700 91800 91900 91900 92000 92100 92200 92300 92400 92500 92600 92700 92800 92900 92900 93000 93100 93200 93300 93400 93500 93600 93700 93800 93900 93900 94000 94100 94200 94300 94400 94500 94600 94700 94800 94900 94900 95000 95100 95200 95300 95400 95500 95600 95700 95800 95900 95900 96000 96100 96200 96300 96400 96500 96600 96700 96800 96900 96900 97000 97100 97200 97300 97400 97500 97600 97700 97800 97900 97900 98000 98100 98200 98300 98400 98500 98600 98700 98800 98900 98900 99000 99100 99200 99300 99400 99500 99600 99700 99800 99900 100000 100100 100200 100300 100400 100500 100600 100700 100800 100900 100900 101000 101100 101200 101300 101400 101500 101600 101700 101800 101900 101900 102000 102100 102200 102300 102400 102500 102600 102700 102800 102900 102900 103000 103100 103200 103300 103400 103500 103600 103700 103800 103900 103900 104000 104100 104200 104300 104400 104500 104600 104700 104800 104900 104900 105000 105100 105200 105300 105400 105500 105600 105700 105800 105900 105900 106000 106100 106200 106300 106400 106500 106600 106700 106800 106900 106900 107000 107100 107200 107300 107400 107500 107600 107700 107800 107900 107900 108000 108100 108200 108300 108400 108500 108600 108700 108800 108900 108900 109000 109100 109200 109300 109400 109500 109600 109700 109800 109900 109900 110000 110100 110200 110300 110400 110500 110600 110700 110800 110900 110900 111000 111100 111200 111300 111400 111500 111600 111700 111800 111900 111900 112000 112100 112200 112300 112400 112500 112600 112700 112800 112900 112900 113000 113100 113200 113300 113400 113500 113600 113700 113800 113900 113900 114000 114100 114200 114300 114400 114500 114600 114700 114800 114900 114900 115000 115100 115200 115300 115400 115500 115600 115700 115800 115900 115900 116000 116100 116200 116300 116400 116500 116600 116700 116800 116900 116900 117000 117100 117200 117300 117400 117500 117600 117700 117800 117900 117900 118000 118100 118200 118300 118400 118500 118600 118700 118800 118900 118900 119000 119100 119200 119300 119400 119500 119600 119700 119800 119800 119900 119900 120000 120100 120200 120300 120400 120500 120600 120700 120800 120900 120900 121000 121100 121200 121300 121400 121500 121600 121700 121800 121900 121900 122000 122100 122200 122300 122400 122500 122600 122700 122800 122900 122900 123000 123100 123200 123300 123400 123500 123600 123700 123800 123900 123900 124000 124100 124200 124300 124400 124500 124600 124700 124800 124900 124900 125000 125100 125200 125300 125400 125500 125600 125700 125800 125900 125900 126000 126100 126200 126300 126400 126500 126600 126700 126800 126900 126900 127000 127100 127200 127300 127400 127500 127600 127700 127800 127900 127900 128000 128100 128200 128300 128400 128500 128600 128700 128800 128900 128900 129000 129100 129200 129300 129400 129500 129600 129700 129800 129800 129900 129900 130000 130100 130200 130300 130400 130500 130600 130700 130800 130900 130900 131000 131100 131200 131300 131400 131500 131600 131700 131800 131900 131900 132000 132100 132200 132300 132400 132500 132600 132700 132800 132900 132900 133000 133100 133200 133300 133400 133500 133600 133700 133800 133900 133900 134000 134100 134200 134300 134400 134500 134600 134700 134800 134900 134900 135000 135100 135200 135300 135400 135500 135600 135700 135800 135900 135900 136000 136100 136200 136300 136400 136500 136600 136700 136800 136900 1369